Script generated by TTT

Title: Petter: Programmiersprachen (25.11.2015)

Date: Wed Nov 25 15:09:56 CET 2015

Duration: 36:43 min

Pages: 20

TECHNISCHE UNIVERSITÄT MÜNCHEN
FAKULTÄT FÜR INFORMATIK

Programming Languages

Dispatching Method Calls

Dr. Michael Petter
Winter Term 2015

Dispatching - Outline

Dispatching

1. Motiviation
2. Formal Model
3. Quiz
4. Dispatching from the Inside

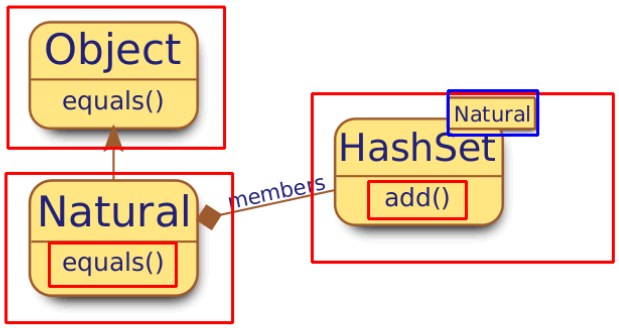
Solutions in Single-Dispatching

1. Type introspection
2. Generic interface

Multi-Dispatching

1. Formal Model
2. Multi-Java
3. Multi-dispatched compare in Java
4. Multi-dispatching in Clojure

Sets of Natural Numbers



Sets of Natural Numbers

```
class Natural {
  Natural(int n){ number=Math.abs(n); }
  int number;
  public boolean equals(Natural n){
    return n.number == number;
  }
}
...
  Set<Natural> set = new HashSet<>();
  set.add(new Natural(0));
  set.add(new Natural(0));
  System.out.println(set);
```


Sets of Natural Numbers

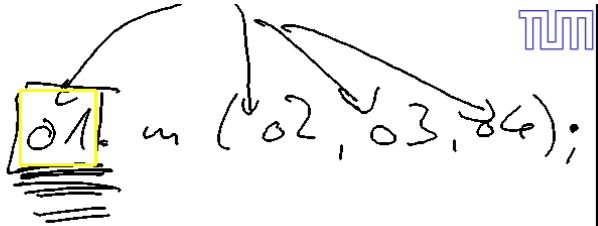
```
class Natural {
  Natural(int n){ number=Math.abs(n); }
  int number;
  public boolean equals(Natural n){
    return n.number == number;
  }
}
...
  Set<Natural> set = new HashSet<>();
  set.add(new Natural(0));
  set.add(new Natural(0));
  System.out.println(set);
```

o1.equals(o2)

```
>$ java Natural
[0,0]
```

⚠ Why? Is HashSet buggy?

Generalization



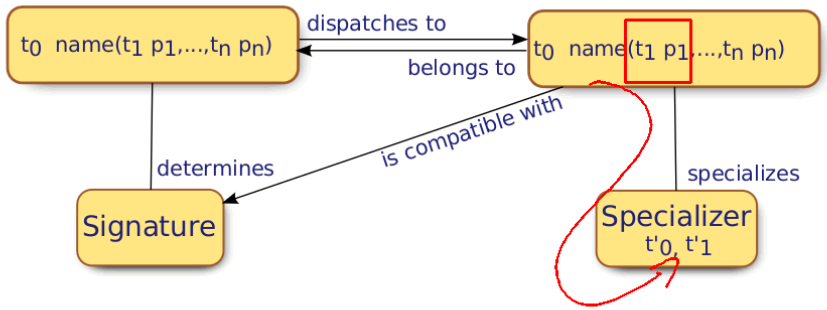
Let's think language independent!

Generalization

Let's think language independent!

`n1.equals(n2);` $\Longrightarrow$ `equals(n1,n2);`

Methods are *dynamically dispatched*



Example: Java [4]

Java determines *generic function signatures* implicitly at each call site from the static types of the arguments.

```
Object o1 = new Natural(1);
Object o2 = new Natural(2);
equals(o1,o2);
```

Signature for call to generic function:

```
equals(Object, Object)
```

Concrete methods within Natural:

```
boolean equals(Natural n1, Natural n2)
boolean equals(Object o1, Object o2)
```


Example: Java [4]

Java determines *generic function signatures* implicitly at each call site from the static types of the arguments.

```
Object o1 = new Natural(1);
Object o2 = new Natural(2);
equals(o1,o2);
```

Signature for call to generic function:

```
equals(Object, Object)
```

⚠ Specializer in Java only for return type and first argument

Concrete methods within Natural:

```
boolean equals(Natural n1, Natural n2)
boolean equals(Object o1, Object o2)
boolean equals(Natural o1, Object o2)
```


Mini-Quiz: Java Method Dispatching

```
class A {
   public static  void p  (Object o) { System.out.println(o); }
   public         void m1 (A a) {      p("m1(A) in A");       }
   public         void m1 () {         m1(new B());           }
   private static void m2 (A a) {      p("m2(A) in A");       }
   public         void m2 () {         m2(this);              }
}
class B extends A {
   public         void m1 (B b) {      p("m1(B) in B");       }
   public         void m2 (A a) {      p("m2(A) in B");       }
   public         void m3 () {         super.m1(this);        }
}
```

```
B b = new B(); A a = b; a.m1(b);
```

Mini-Quiz: Java Method Dispatching

```
class A {
   public static  void p  (Object o) { System.out.println(o); }
   public         void m1 (A a) {      p("m1(A) in A");       }
   public         void m1 () {         m1(new B());           }
   private static void m2 (A a) {      p("m2(A) in A");       }
   public         void m2 () {         m2(this);              }
}
class B extends A {
   public         void m1 (B b) {      p("m1(B) in B");       }
   public         void m2 (A a) {      p("m2(A) in B");       }
   public         void m3 () {         super.m1(this);        }
}
```

```
B b = new B(); A a = b; a.m1(b);
```

m1(A) in A

Mini-Quiz: Java Method Dispatching


```
class A {
   public static  void p  (Object o) { System.out.println(o); }
   public         void m1 (A a) {      p("m1(A) in A");       }
   public         void m1 () {         m1(new B());           }
   private static void m2 (A a) {      p("m2(A) in A");       }
   public         void m2 () {         m2(this);              }
}
class B extends A {
   public         void m1 (B b) {      p("m1(B) in B");       }
   public         void m2 (A a) {      p("m2(A) in B");       }
   public         void m3 () {         super.m1(this);        }
}
```


```
B b = new B(); A a = b; a.m1(b);
```

m1(A) in A

```
B b = new B(); B a = b; b.m1(a);
```

m1(B) in B

Mini-Quiz: Java Method Dispatching


```
class A {
   public static  void p  (Object o) { System.out.println(o); }
   public         void m1 (A a) {      p("m1(A) in A");       }
   public         void m1 () {         m1(new B());           }
   private static void m2 (A a) {      p("m2(A) in A");       }
   public         void m2 () {         m2(this);              }
}
class B extends A {
   public         void m1 (B b) {      p("m1(B) in B");       }
   public         void m2 (A a) {      p("m2(A) in B");       }
   public         void m3 () {         super.m1(this);        }
}
```


```
B b = new B(); A a = b; a.m1(b);
```

m1(A) in A

```
B b = new B(); B a = b; b.m1(a);
```

m1(B) in B

Mini-Quiz: Java Method Dispatching


```
class A {
   public static  void p  (Object o) { System.out.println(o); }
   public         void m1 (A a) {      p("m1(A) in A");       }
   public         void m1 () {         m1(new B());           }
   private static void m2 (A a) {      p("m2(A) in A");       }
   public         void m2 () {         m2(this);              }
}
class B extends A {
   public         void m1 (B b) {      p("m1(B) in B");       }
   public         void m2 (A a) {      p("m2(A) in B");       }
   public         void m3 () {         super.m1(this);        }
}
```


```
B b = new B(); A a = b; a.m1(b);
```

m1(A) in A

```
B b = new B(); B a = b; b.m1(a);
```

m1(B) in B

```
B b = new B(); b.m1();
```

Mini-Quiz: Java Method Dispatching

```
class A {
   public static  void p  (Object o) { System.out.println(o); }
   public         void m1 (A a) {      p("m1(A) in A");       }
   public         void m1 () {         m1(new B());           }
   private static void m2 (A a) {      p("m2(A) in A");       }
   public         void m2 () {         m2(this);              }
}
class B extends A {
   public         void m1 (B b) {      p("m1(B) in B");       }
   public         void m2 (A a) {      p("m2(A) in B");       }
   public         void m3 () {         super.m1(this);        }
}
```

```
B b = new B(); A a = b; a.m1(b);                    m1(A) in A
B b = new B(); B a = b; b.m1(a);                    m1(B) in B
B b = new B(); b.m1();                              m1(A) in A
```


Mini-Quiz: Java Method Dispatching


```
class A {
   public static  void p  (Object o) { System.out.println(o); }
   public         void m1 (A a) {      p("m1(A) in A");       }
   public         void m1 () {         m1(new B());           }
   private static void m2 (A a) {      p("m2(A) in A");       }
   public         void m2 () {         m2(this);              }
}
class B extends A {
   public         void m1 (B b) {      p("m1(B) in B");       }
   public         void m2 (A a) {      p("m2(A) in B");       }
   public         void m3 () {         super.m1(this);        }
}
```

```
B b = new B(); A a = b; a.m1(b);                    m1(A) in A
B b = new B(); B a = b; b.m1(a);                    m1(B) in B
B b = new B(); b.m1();                              m1(A) in A
```


```
B b = new B(); b.m2();
```

Dispatching Method Calls | The Problem | Formal | 8 / 31

Mini-Quiz: Java Method Dispatching


```
class A {
   public static  void p  (Object o) { System.out.println(o); }
   public         void m1 (A a) {      p("m1(A) in A");       }
   public         void m1 () {         m1(new B());           }
   private static void m2 (A a) {      p("m2(A) in A");       }
   public         void m2 () {         m2(this);              }
}
class B extends A {
   public         void m1 (B b) {      p("m1(B) in B");       }
   public         void m2 (A a) {      p("m2(A) in B");       }
   public         void m3 () {         super.m1(this);        }
}
```

```
B b = new B(); A a = b; a.m1(b);                    m1(A) in A
B b = new B(); B a = b; b.m1(a);                    m1(B) in B
B b = new B(); b.m1();                              m1(A) in A
B b = new B(); b.m2();                              m2(A) in A
```


Dispatching Method Calls | The Problem | Formal | 8 / 31

Mini-Quiz: Java Method Dispatching


```
class A {
   public static  void p  (Object o) { System.out.println(o); }
   public         void m1 (A a) {      p("m1(A) in A");       }
   public         void m1 () {         m1(new B());           }
   private static void m2 (A a) {      p("m2(A) in A");       }
   public         void m2 () {         m2(this);              }
}
class B extends A {
   public         void m1 (B b) {      p("m1(B) in B");       }
   public         void m2 (A a) {      p("m2(A) in B");       }
   public         void m3 () {         super.m1(this);        }
}
```

```
B b = new B(); A a = b; a.m1(b);                    m1(A) in A
B b = new B(); B a = b; b.m1(a);                    m1(B) in B
B b = new B(); b.m1();                              m1(A) in A
B b = new B(); b.m2();                              m2(A) in A
B b = new B(); b.m3();                              m1(A) in A
```


Dispatching Method Calls | The Problem | Formal | 8 / 31

So what is happening here?

Let's look at what Java does!

The Java platform as example for state of the art OO systems:

- Static Javac-based compiler
- Dynamic Hotspot JIT-Compiler/Interpreter

Let's watch the following code on its way to the CPU:

```
public static void main(String[] args){
    Object o1 = new Natural(1);
    Object o2 = new Natural(2);
    o1.equals(o2);
}
```